Chapter I:

Software Requirements Specification

DRAFT 4/97

1.0 INTRODUCTION

1.1 Purpose

This chapter, "Software Requirements Specification," documents the functional and performance requirements for the FEHM application. It shall be used in designing and implementing the requirements for the application and shall be employed in subsequent baselines to verify that the application has fully implemented these requirements.

1.2 Scope

The FEHM application is based on a finite-element heat- and mass-transfer code that simulates nonisothermal, multiphase, multicomponent flow, and solute transport in porous media. It is applicable to natural-state studies of geothermal systems and groundwater flow. The FEHM code will be used for parameter sensitivity studies in the design and specification of field tracer and flow experiments and the interpretation of those field experiments. In addition, it will be used for field-scale simulations of radionuclide migration in the saturated and unsaturated zones below Yucca Mountain.

2.0 GENERAL DESCRIPTION

2.1 Software Perspective

FEHMN is a version of FEHM (Zyvoloski et al. 1988) that was developed for the C-wells project in the mid 1980s when LANL was analyzing the early well test data of the USGS. Most of the development of FEHM was done while one of the authors (Zyvoloski) was a participant of the Los Alamos Hot Dry Rock Project.

FEHM and FEHMN are aliases that represent the same computer code, and hence forth, the code will be referenced by the simpler acronym, FEHM.

2.2 Software Functions

The major software functions of FEHM are:

- calculate finite-element coefficients for a mesh that represents the porous and permeable medium;
- formulate the transient equations for heat conduction, heat and mass transfer for multiphase flow within porous and permeable media, noncondensible gas flow within porous and permeable media, and transport of multiple solutes within porous and permeable media;
- apply constitutive relationships for pressure- and temperature-dependent fluid properties, air/gas properties, relative permeabilities and capillary pressures, reactive and sorbing solutes, dual porosity, double porosity/double permeability, and stress-dependent properties;
- compute the finite-element solution to the transient equations;
- represent the simulation input and output results in a suite of data files;
- perform a restart calculation in which the progress of the flow field is simulated starting at the time and conditions at which a previous run terminated.

2.3 User Characteristics

The user should have experience in reservoir engineering, hydrology, or porous media flow. The user should also have experience with numerical simulations.

DRAFT 4/97

2.4 General Constraints

Problem size and limits are dictated by the capabilities of the computer on which the simulation is run. The code shall be able to handle extreme property changes (e.g., large changes in permeability between adjacent strata or highly nonlinear characteristic curves) and systems containing both saturated and unsaturated regions. Provided that the user supplies physically meaningful parameter values, does not violate model assumptions, and runs problems of the type presented in the verification report, results should be reliable.

2.5 Assumptions and Dependencies

FEHM will be written in Fortran 77. It will be developed to run on CRAY and SUN computers. (FEHM has been successfully run on CRAY, SUN, VAX, HP (Unix-based) and IBM RISC computers.) FEHM requires 64-bit precision.

3.0 FUNCTIONAL REQUIREMENTS

Each section below in this chapter deals with a major function enumerated in Section 2.2 above. After a user formulates a problem to be modeled, a mesh representing the physical media would be generated. Data describing the finite-element mesh would be input (Section 3.5) and the element coefficients computed (Section 3.1). Data describing the physical properties of the media and the fluid/gas system to be modeled would be input (Section 3.5). The appropriate finite-element equations would be formulated (Section 3.2) and constitutive relationships applied (Section 3.3). The solution would then be computed (Section 3.4) and results and restart data output (Sections 3.5 and 3.6).

3.1 Finite-element Coefficient Generation

3.1.1 Introduction

The application shall include routines for the generation of finite-element coefficients. These coefficients represent the purely geometric part of the numerical analog of the partial differential equations that describe the coupled flow processes in porous media. These coefficients include the volume associated with each grid point as well as an area/distance term for all internode pairs. In standard finite-element terminology, this process results in the geometric components of the stiffness matrix.

3.1.2 Inputs

The following data are required to generate the finite-element coefficients:

- the problem dimension (2 or 3 dimensions) and geometry;
- the number of nodes and elements used to define the solution space;
- the node number and x-, y-, and z-coordinates of each node in the finite-element mesh;
- the number of nodes per element (3-noded triangles or 4-noded quadrilaterals for 2-D problems and 4-noded tetrahedrals, 6-noded triangular prisms, or 8-noded quadrilateral polyhedrons for 3-D problems); and
- the element number and nodal connectivity (which nodes define the element) for each element using a right-handed coordinate system.

3.1.3 Processing

The calculation of the finite-element coefficients is accomplished by using the coordinate and element data (type and node list) to develop the

connectivity of the node points and volumes and areas associated with the node points. These calculations are first done on an element level and then combined on a node level.

3.1.4 Outputs

Data structures containing the finite-element coefficients for the input mesh shall be output. The volume of each grid point is contained in an array indexed by node number. The area/distance terms are in arrays indexed by the connectivity structure of the grid. This structure is the same as that containing the neighboring nodes of the grid points.

3.2 Formulate Transient Equations

The transient equations shall be formulated, and the system of equations derived from the finite-element representation of the overall mathematical model shall be solved.

3.2.1 Heat-conduction equations

3.2.1.1 Introduction

FEHM shall be capable of simulating the conduction of heat without the flow of any fluids. The basic governing equation is

$$-\overline{\nabla} \cdot (K\overline{\nabla}T) + q + \frac{\partial A_e}{\partial t} = 0 \quad , \tag{1}$$

where K is an effective thermal conductivity, T is the temperature, A_e is the energy per unit volume, and q is the specified energy source term.

3.2.1.2 Inputs

All property and parameter values implied by Eqn. (1) are required as inputs: density, enthalpy, specific heat, thermal conductivity, initial temperatures, boundary values, coordinates of the computational mesh points, and element definitions.

3.2.1.3 Processing

The time derivatives in Eqn. (1) shall be discretized using the standard first-order method (Hinton and Owen 1979) given by

$$f(t^{n+1}) = f(t^n) + \Delta t[a f'(t^{n+1}) + (1-a) f'(t^n)] \quad , \tag{2}$$

where $f(t^{n+1})$ is the desired function at time t^{n+1}, $f(t^n)$ is the known value of f at time t^n, Δt is the time step, f' is the derivative of f with respect to time, and a is a weighting factor. The space derivatives in the governing equation shall be discretized using the finite-element formulation (Hinton and Owen 1979). Using numerical integration, the shape-function coefficients for the discretized equations shall be generated at each node.

DRAFT 4/97

3.2.1.4 Outputs

The outputs shall be the equation coefficients for the finite-element formulation of the governing equation (Eqn. (1)) at each node.

3.2.2 Heat- and mass-transfer equations

3.2.2.1 Introduction

FEHM shall also be capable of simulating the transport of heat and mass within porous and permeable media. The conservation equations for heat and mass transfer are

$$-\overline{\nabla}\cdot(D_{mv}\overline{\nabla}P_v) - \overline{\nabla}\cdot(D_{ml}\overline{\nabla}P_l) + q_m + \frac{\partial}{\partial z}g(D_{mv}\rho_v + D_{ml}\rho_l) + \frac{\partial A_m}{\partial t} = 0 \qquad (3)$$

and

$$-\overline{\nabla}\cdot(D_{ev}\overline{\nabla}P_v) - \overline{\nabla}\cdot(D_{el}\overline{\nabla}P_l) - \overline{\nabla}\cdot(K\overline{\nabla}T) + q_e + \frac{\partial}{\partial z}g(D_{ev}\rho_v + D_{el}\rho_l) + \frac{\partial A_e}{\partial t} = 0\ , \qquad (4)$$

where D is the transmissibility, K is an effective thermal conductivity, P is the pressure, T is the temperature, A_m is the mass per unit volume, A_e is the energy per unit volume, q represents the source and sink terms (such as bores, reinjection wells, or groundwater recharge), ρ is the density, z is oriented in the direction of gravity, and g represents the acceleration due to gravity. Here also, the subscripts v and l indicate quantities for the vapor phase and the liquid phase, respectively, and the subscripts m and e refer to mass and energy, respectively.

3.2.2.2 Inputs

The inputs required are all property and parameter values implied by Eqns. (3) and (4): permeabilities, porosities, viscosities, densities, internal energies, specific heats, enthalpies, thermal conductivities, initial pressures, initial temperatures, initial saturations, boundary values, coordinates of the computational mesh points, and element definitions.

3.2.2.3 Processing

The time derivatives in Eqns. (3) and (4) shall be discretized using Eqn. (2). The space derivatives in the governing equations shall be discretized using the finite-element formulation (Hinton and Owen 1979). Using numerical integration, the shape-function coefficients for the discretized equations shall be generated at each node.

DRAFT 4/97

3.2.2.4 Outputs

The outputs shall be the equation coefficients for the finite-element formulation of the governing equations (Eqns. (3) and (4)) at each node.

3.2.3 Noncondensible gas flow equations

3.2.3.1 Introduction

In addition to the flow of heat and mass, FEHM shall also be capable of simulating noncondensible gas flow (usually air). The noncondensible gas conservation equation is given by

$$-\overline{\nabla}\cdot(C_v D_{mv}\overline{\nabla}P_v) - \overline{\nabla}\cdot(C_l D_{ml}\overline{\nabla}P_l) + q_c + \frac{\partial}{\partial z}g(C_v D_{mv}\rho_v + C_l D_{ml}\rho_l) + \frac{\partial A_c}{\partial t} = 0 . \quad (5)$$

Here, C is the concentration of the noncondensible gas, expressed as a fraction of total mass, q_c is the source (or sink) strength, and A_c is the accumulation term.

3.2.3.2 Inputs

The inputs required are all property and parameter values implied by Eqn. (5): gas concentrations, permeabilities, porosities, viscosities, densities, initial pressures and saturations, boundary values, coordinates of the computational mesh points, and element definitions.

3.2.3.3 Processing

The time derivatives in Eqn. (5) shall be discretized using Eqn. (2). The space derivatives in the governing equation shall be discretized using the finite-element formulation (Hinton and Owen 1979). Using numerical integration, the shape-function coefficients for the discretized equations shall be generated at each node.

3.2.3.4 Outputs

The outputs will be the equation coefficients for the finite-element formulation of the governing equation (Eqn. (5)) at each node.

3.2.4 Solute-transport equations

3.2.4.1 Introduction

FEHM shall also be capable of simulating solute transport and shall allow for simultaneous solution of multiple, interacting solutes. The passive solute equations are not directly coupled to the pressure field but use the pressure field obtained by the heat- and mass-transfer solution. The transport equation for a given component is given by

DRAFT 4/97

$$-\overline{\nabla}\cdot(C_v D_{mv}\overline{\nabla}P_v) - \overline{\nabla}\cdot(C_l D_{ml}\overline{\nabla}P_l) - \overline{\nabla}\cdot(D_c\overline{\nabla}C_l) +$$

$$q_c + \frac{\partial}{\partial z}g(C_v D_{mv}\rho_v + C_l D_{ml}\rho_l) + \rho_r\frac{\partial C_R}{\partial t} + \frac{\partial A_c}{\partial t} = 0 \quad . \quad (6)$$

Here, C is the concentration of the solute. The additional terms from those in Eqn. (5) are $\overline{\nabla}\cdot(D_c\overline{\nabla}C_l)$, the dispersion term, and $\rho_r\frac{\partial C_R}{\partial t}$, the adsorption term. C_R represents the adsorption of species onto the porous media. In addition, the term q_c includes the source or sink due to chemical reaction. The formulation for multiple interacting species used to compute this term is outlined in Section 3.3.6.

3.2.4.2 Inputs

The required inputs are all property and parameter values implied by Eqn. (6): solute concentrations, permeabilities, porosities, viscosities, densities, initial pressures and saturations, boundary values, coordinates of the computational mesh points, and element definitions.

3.2.4.3 Processing

The time derivatives in Eqn. (6) shall be discretized using Eqn. (2). The space derivatives in the governing equation shall be discretized using the finite-element formulation (Hinton and Owen 1979). Using numerical integration, the shape-function coefficients for the discretized equations shall be generated at each node.

3.2.4.4 Outputs

The outputs shall be the equation coefficients for the finite-element formulation of the governing equation (Eqn. (6)) at each node.

3.2.5 Particle-tracking module

3.2.5.1 Introduction

FEHM shall be capable of simulating solute transport in either the liquid or gas phases using a cell-based particle-tracking algorithm. The algorithm shall be developed to handle either steady-state or transient flow fields for either an equivalent continuum or dual-permeability flow field. The algorithm shall include the following transport submodels:

- dispersion, to be simulated by applying a correction to the time of a particle in a cell based on a transfer function described by a solution to the one-dimensional advective-dispersion equation;
- sorption, using an equilibrium, linear sorption or "K_d" model;

- matrix diffusion with sorption on either the fracture faces or the matrix rock, implemented using the transfer-function approach similar to the method described above for dispersion, with a one-dimensional matrix-diffusion submodel; and
- radioactive decay.

3.2.5.2 Inputs

The required inputs shall be:

- internode mass fluxes and the fluid storage at each node;
- the injection history for particles; and
- transport parameters for the solute, including sorption coefficients, radioactive decay constants, dispersion coefficients, and diffusion coefficients.

3.2.5.3 Processing

Using the inputs listed above, the code shall track the progress of particles from node to node subject to the transport processes of diffusion, dispersion, and sorption. At specified times, the code shall supply the particle concentrations at each node, and at the final time of the simulation, the code shall output the information on each particle such that a restart of the calculation can be performed.

3.2.5.4 Outputs

The outputs shall be the time-varying locations and concentrations of particles.

3.2.6 Sources and sinks

3.2.6.1 Introduction

The following types of sources and sinks shall be implemented:

- specified flow rate;
- specified pressure condition;
- specified heat flux;
- specified temperature;
- specified flux and saturation;
- specified water flux and air pressure;
- specified air flux (no water flow);
- specified water flux (no air flow);
- specified relative humidity of the gas phase; and
- specified air pressure and saturation.

3.2.6.2 Inputs

For each source/sink (depending on source/sink type), the required inputs are:

- source/sink location;
- source/sink strength;
- flowing pressure;

- enthalpy or temperature of the fluid injected;
- species concentration in the fluid injected;
- impedance;
- thermal resistance;
- saturation;
- relative humidity; and
- air pressure.

3.2.6.3 Processing

Processing shall consist of determine the pressure, mass flow rate, heat flow, species concentration, saturation, or air flux at each source/sink.

3.2.6.4 Outputs

The outputs shall be pressure, mass flow rate, heat flow, species concentration, saturation, or air flow at each source/sink.

3.3 Apply Constitutive Relationships

The following constitutive relations shall be available.

3.3.1 Pressure- and temperature-dependent water properties

3.3.1.1 Introduction

In the governing equations described above (Eqns. (1), (3), (4), (5), and (6)), physical properties such as the density, enthalpy, and viscosity can be strong functions of pressure (P) and temperature (T) making the governing equations very nonlinear. Rational-function approximations (Zyvoloski and Dash 1991b) shall be used to estimate the thermodynamic variables in FEHM over the ranges $0.001 \le P \le 110.0$ MPa and $0.001 \le T \le 360$°C. In addition, a second set of rational-function approximations shall be provided for low pressures to give better estimates of the thermodynamic variables over the ranges $0.001 \le P \le 20.0$ MPa and $0.5 \le T \le 360$°C.

3.3.1.2 Inputs

The required inputs shall be pressure and temperature at each node at each time step.

3.3.1.3 Processing

The processing shall compute the water properties and their derivatives, using the rational-function approximations, as a function of pressure and temperature at each node at each time step.

3.3.1.4 Outputs

The outputs shall be water density, enthalpy, and viscosity (and their derivatives) at each node at each time step.

DRAFT 4/97

3.3.2 Properties of air and air/water vapor mixtures

3.3.2.1 Introduction

The flow of air and air/water vapor mixtures in the porous media shall also be modeled. Appropriate thermodynamic information (density, enthalpy, viscosity, mass fractions) for air and air/water vapor mixtures shall be provided.

3.3.2.2 Inputs

The required inputs will be pressure, temperature, and air/vapor/liquid mass fractions at each node at each time step.

3.3.2.3 Processing

The processing will compute the air and mixture properties at each node at each time step.

3.3.2.4 Outputs

The outputs shall be air and mixture properties (densities, enthalpies, and viscosities) at each node at each time step.

3.3.3 Equation-of-state models

3.3.3.1 Introduction

The ability to use liquids other than water and to use gases other than air in the porous media model shall be provided. The use of equation-of-state models with user-defined parameters allows the introduction of alternate fluids, ideal gases, etc. The models can also be used to provide simplified state relations for water and air.

3.3.3.2 Inputs

The required inputs shall be liquid and vapor reference pressure, temperature, density, enthalpy, and viscosity and their derivatives with respect to temperature and pressure at the reference conditions.

3.3.3.3 Processing

The processing shall compute the fluid properties based on equation-of-state data at each node at each time step.

3.3.3.4 Outputs

The outputs shall be fluid properties (densities, enthalpies, and viscosities) at each node at each time step.

3.3.4 Relative-permeability and capillary-pressure functions

3.3.4.1 Introduction

Relative permeabilities and capillary pressures can be strong functions of saturation. Several well-known relative-permeability functions shall be made available to the user. These functions are the simple linear functions, the Corey (1954) relationships, and the van Genuchten (1980) functions. Composite relative-permeability curves, as described by Klavetter and Peters (1986), shall also be a user option. The

capillary functions considered are the linear function and the van Genuchten (1980) capillary-pressure model.

3.3.4.2 Inputs

The required inputs are saturations, pressures, and model parameters implied by the constitutive model to be used.

3.3.4.3 Processing

The processing shall compute relative permeabilities and capillary pressures based on the selected constitutive model.

3.3.4.4 Outputs

The outputs shall be relative permeabilities and capillary pressures at each node.

3.3.5 Adsorbing solutes

3.3.5.1 Introduction

FEHM shall have provisions for adsorbing solutes. The general equilibrium model for adsorption of species onto the reservoir rock is given by (Polzer et al. 1992):

$$C_R = \frac{\alpha_1 C_l^{\beta}}{1 + \alpha_2 C_l^{\beta}} \cdot \qquad (7)$$

The parameters α_1, α_2, and β are associated with the sorption models.

3.3.5.2 Inputs

The required inputs shall be all property and parameter values implied by Eqn. (7): concentrations and sorption parameters.

3.3.5.3 Processing

The processing will compute adsorption based on the appropriate model and compute the modified coefficients of the tracer transport equation that account for adsorption reactions.

3.3.5.4 Outputs

The outputs shall be equation coefficients for the finite-element formulation of the tracer transport equations at each node modified to account for adsorption reactions.

3.3.6 Multiple, interacting solutes

3.3.6.1 Introduction

FEHM shall have provisions for multiple, interacting solutes in which chemical reactions involving one or more components are specified. Any number of reactions may be specified with the following form:

$$a_1 S_1 + a_2 S_2 + \ldots + a_m S_m =$$

$$a_{m+1} S_{m+1} + a_{m+2} S_{m+2} + \ldots + a_n S_n \ , \qquad (8)$$

where the a's are the stoichiometric coefficients and the S's denote each solute present in a particular reaction. This relationship is formulated for each reaction being modeled, and a solute may be present in any number of reactions as either a reactant or a product.

The rate law governing each reversible reaction shall be specified as follows:

$$\frac{\partial}{\partial t}[S_j] = \pm a_j \left\{ k_{for} \prod_{i=1}^{m} [S_i]^{b(i)} - k_{rev} \prod_{i=m+1}^{n} [S_i]^{b(i)} \right\} \ , \qquad (9)$$

where the square brackets [ ] denote concentration, the $b(i)$ are exponents in the reaction rate equation (specified for every reactant in each reaction), and the forward and reverse reaction rate constants, k_{for} and k_{rev}, are governed by the Arrhenius equation. An option shall be provided to allow or disallow chemical reaction for solute residing on the rock for the case of a sorbing component. Finally, an input option shall be developed so that the user may specify an equilibrium constant for the chemical reaction with the code generating the rate constants k_{for} and k_{rev} such that equilibrium is attained at each time step.

3.3.6.2 Inputs

The required inputs are all property and parameter values implied by Eqns. (8) and (9): concentrations and chemical-reaction model parameters.

3.3.6.3 Processing

The processing shall compute the chemical-reaction source/sink term for each solute.

3.3.6.4 Outputs

The outputs shall be equation coefficients for the finite-element formulation of the tracer transport equations at each node modified to account for chemical reactions.

3.3.7 Dual-porosity formulation

3.3.7.1 Introduction

Many porous-medium problems are dominated by fracture flow. In these cases, the fracture permeability controls the pressure communication in the reservoir even though local storage around the fracture may be dominated by the porous rock, which communicates only with the closest fractures.

Computationally, a volume fraction and length scale shall be used to create one-dimensional versions of Eqns. (3), (4), (5), and (6). The length scale shall be used to modify spatial difference terms, and the volume factors shall be used to modify the accumulation terms. The volume fractions for the dual-porosity formulation shall satisfy the following relationship:

$$V_f + V_{f1} + V_{f2} = 1 \ , \tag{10}$$

where V_f is the volume fraction of fractures, V_{f1} is the fraction of the first matrix volume, and V_{f2} is the fraction of the second matrix volume. Note that two nodes shall be used to model the porous rock (matrix). The length scales shall be given by

$$\begin{aligned} L_f &= L_{f0} V_f \\ L_{f1} &= L_{f0} V_f \\ L_{f2} &= L_{f0} V_f \end{aligned} \ , \tag{11}$$

where L_f is the length scale for the fracture volume, L_{f1} is the length scale of the first matrix volume, L_{f2} is the length scale of the second matrix volume, and L_{f0} is a characteristic length scale.

3.3.7.2 Inputs

The required inputs shall be:

- volume fraction of the fracture (V_f) and matrix volumes (V_{f1}, V_{f2}) and
- characteristic length scale for fracture spacing (L_{f0}).

3.3.7.3 Processing

The processing shall modify the equation coefficients (from the finite-element formulation of Eqns. (3), (4), (5), and (6)) at each node using the volume fractions and length scales.

3.3.7.4 Outputs

The outputs shall be the modified equation coefficients at each dual-porosity node.

3.3.8 Double-porosity/double-permeability formulation

3.3.8.1 Introduction

In a partially saturated porous medium, flow is often dominated by capillary suction. In a medium comprised of fractures and matrix, the matrix material has the highest capillary suction, and under relatively static conditions, the moisture resides in the matrix material. Infiltration events, such as severe rainfall, can saturate the porous medium allowing rapid flow in the fractures. To capture this flow phenomena, a system of equations allowing communication between the fractures and

matrix blocks in the reservoir, in addition to the flow within the fractures and matrix blocks, is necessary.

A volume fraction and length scale (different from the dual-porosity formulation) shall be used to characterize the system. Equations (3), (4), (5), and (6) are formulated for both the fracture and matrix computational grids. One-dimensional versions are created to couple locally the two sets of equations. As in the dual-porosity formulation, matrix material communicates only with the local fractures.

The volume fractions for the double-porosity/double-permeability formulation shall satisfy the following relationship:

$$V_f + V_{f1} = 1 \quad , \tag{12}$$

where V_f is the volume fraction of fractures and V_{f1} is the fraction of the matrix volume. The length scales shall be partitioned for the fracture and matrix volumes using

$$\begin{aligned} L_f &= L_{f0} V_f \\ L_{f1} &= L_{f0} V_{f1} \end{aligned} \quad , \tag{13}$$

where L_f is the length scale for the fracture volume, L_{f1} is the length scale of the matrix volume, and L_{f0} is a characteristic length scale.

3.3.8.2 Inputs

The required inputs shall be:

- volume fraction of the fracture (V_f) and
- characteristic length scale for fracture spacing (L_{f0}).

3.3.8.3 Processing

The processing shall modify the equation coefficients (from the finite-element formulation of Eqns. (3), (4), (5), and (6)) at each node using the volume fractions and length scales.

3.3.8.4 Outputs

The outputs shall be the modified equation coefficients at each double-porosity/double-permeability node.

3.3.9 Stress-dependent properties

3.3.9.1 Introduction

Often, it is necessary to accommodate changes in the rock porosity and permeability due to changes in effective stress caused by temperature and pore-fluid pressure changes. A linear and nonlinear model shall be incorporated in the code for this purpose.

The linear pore-pressure model for porosity is given by

$$\phi = \phi_0 + (1-\phi_0)(c_r - c_g)(P - P_0) \quad , \tag{14}$$

where ϕ is the porosity at pressure P, ϕ_0 is the porosity at pressure P_0, c_r is the pore volume compressibility of the rock, and c_g is the compressibility of the matrix grain material.

The nonlinear model for porosity (Gangi 1978) is given by

$$\phi = \phi_0\left[1 - \left(\frac{P_c}{P_x}\right)^m\right] \tag{15}$$

and

$$P_c = \sigma - P - \alpha E \Delta T \ , \tag{16}$$

where P_c is the closure stress, σ is the *in situ* stress (assumed isotropic), α is the coefficient of thermal expansion of the rock, E is Young's modulus, ΔT is the temperature change of the rock, and P_x and m are parameters in the model.

For either case, the effect of stress and temperature changes on permeability shall be modeled with

$$k = k_0\left(\frac{\phi}{\phi_0}\right)^3 \ , \tag{17}$$

where k is the permeability at porosity ϕ.

3.3.9.2 Inputs

The required inputs shall be the parameters of the appropriate pressure- and temperature-dependent matrix-properties model: pressure, porosity, compressibility, closure stress, *in situ* stress, coefficient of thermal expansion of the rock, Young's modulus, temperature change of the rock, and permeability.

3.3.9.3 Processing

The processing shall compute the fracture and matrix property values using the appropriate pressure- and temperature-dependent matrix-properties model at each node for porous-media, dual-porosity, or double-porosity/double-permeability formulations.

3.3.9.4 Outputs

The outputs shall be the porosity and permeability of each matrix cell.

DRAFT 4/97

3.3.10 Variable thermal conductivity

3.3.10.1 Introduction

The thermal conductivity of the solid is often more accurately characterized as a function of temperature or liquid saturation. A linear temperature-dependent model and a relation based upon the square root of liquid saturation shall be incorporated in the code for this reason.

The linear temperature-dependent model is given by

$$K_T = K_{ref} + K_s(T - T_{ref}) \ , \tag{18}$$

where K_T is the temperature-dependent thermal conductivity, K_{ref} is the thermal conductivity at the reference temperature T_{ref}, and K_s is the slope of the linear relation.

The saturation-dependent thermal-conductivity model is given by

$$K_{sat} = K_{dry} + K_{s,s}\sqrt{S} \ , \tag{19}$$

where K_{sat} is the saturation-dependent thermal conductivity, K_{dry} is the conductivity at zero saturation, and $K_{s,s}$ is the slope of the linear relationship. Note that the conductivity at complete saturation is $K_{dry} + K_{s,s}$.

3.3.10.2 Inputs

The required inputs shall be the parameters of the appropriate saturation or temperature-dependent thermal conductivity model: temperature, liquid saturation, and the parameters in the above equations.

3.3.10.3 Processing

The processing shall compute the thermal conductivity based on the above relations.

3.3.10.4 Outputs

The output shall be the rock thermal conductivity at each cell.

3.4 Compute Solution to Transient Equations

3.4.1 Implement time-step mechanism

3.4.1.1 Introduction

Transient finite-element solutions proceed forward in time, computing the solution at each time step. A mechanism shall be implemented in which the user can control the initial size of the time step, the maximum allowable time step, and the rate of

increase of the time step from one time to the next. This mechanism shall also allow the user to reinitialize the time step, if desired, during the simulation (e.g., a pressure at a source/sink changes dramatically and increased detail is desired or needed for convergence). The time-step mechanism shall also be developed so that the time step is automatically reduced if the solution at a time step does not converge within a specified number of iterations.

3.4.1.2 Inputs

The required inputs shall be:

- the initial time-step size;
- the minimum and maximum time-step values;
- the multiplication factor for increasing the size of the time step;
- a data structure specifying when to set the time step to a new value, what that value is, and what to use for a new multiplication factor;
- the maximum number of iterations to be taken within a time step before the time step is reduced; and
- the maximum number of iterations to be taken for the solution of the total problem.

3.4.1.3 Processing

The processing shall determine the new time step after convergence is achieved for the solution at the current time, and this time step will be added to the current time to obtain the new time. The time step shall not be increased above the maximum specified value. If convergence is not achieved, the time step will be reduced and the step repeated. The time step shall not be reduced below the minimum time-step value.

3.4.1.4 Outputs

The outputs shall be the new time step, the new time, and the number of iterations used to achieve convergence. If the time step has been reduced to the minimum time-step value and convergence is not achieved or the maximum number of iterations for the total problem has been reached and convergence is not achieved, an error message shall be issued and the program shall exit.

3.4.2 Solve nonlinear equation set at each time step

3.4.2.1 Introduction

In general, the result of the finite-element formulation (Zienkiewicz 1977) of the transient problem shall be a set of nonlinear algebraic equations with unknown values such as pressure, temperature, and concentration that shall be solved at each time step. The equations shall be solved with a Newton-Raphson iterative-solution scheme.

DRAFT 4/97

3.4.2.2 Inputs

The required inputs shall be:

- the residual equation for each unknown;
- the derivative of each residual with respect to each unknown; and
- the convergence criterion for solving the system of equations.

3.4.2.3 Processing

The processing shall iteratively determine the pressure/temperature/concentration (solute or air) field by solving the nonlinear equation set using the Newton-Raphson technique until a solution is obtained to the desired degree of accuracy. At each iteration, the system of linear algebraic equations shall be solved using the GZSOLVE application (Zyvoloski and Robinson 1995).

For simulations involving solute transport, it shall be possible to disable the heat and mass portion of the solution and just solve the solute-transport equations.

3.4.2.4 Outputs

The outputs shall be the pressure/temperature/concentration field at the new time.

3.5 Provide Input/Output Data Files

3.5.1 Introduction

All input necessary to perform the finite-element simulation of the model shall be read from one or more input data files. The input data shall be organized in groups of related parameters. The information produced by the program shall be written to one or more output data files so that it may be read by programs using the data, such as plotting or data-analysis routines.

3.5.2 Inputs

The required inputs shall be:

- the finite-element mesh data;
- the initial pressures, temperatures, saturations, and concentrations (solute or gas) within the model domain;
- rock property values within the model domain (conductivity, heat capacity, density);
- data used to set heat- and mass-source/sink strength or pressure;
- data used to set concentration (solute or gas) source strength;
- properties of the finite-element mesh representing the matrix (porosity, permeability);
- finite-element parameters used to control the accuracy of the solution, such as the initial and maximum time steps, the time-step multiplication factor, and convergence criterion; and
- model parameters for selected constitutive relations.

DRAFT 4/97

3.5.3 Processing

The processing shall read in all parameter values and generate the finite-element solution to the model. It shall also write the solution to output data files.

3.5.4 Outputs

The outputs shall be:

- pressure, temperature, concentration, and flow field at selected intervals and final time;
- geometric information needed to view the pressure, temperature, concentration, or flow field (i.e., coordinate locations for all nodes); and
- time-varying values of the pressures, temperatures, concentrations, and flow rates of each source/sink or specified matrix location.

3.6 Provide Restart Capability

In large-scale transient simulations, it is often desirable to resume a computation starting at the final time of a previous run or to start with a previously established flow field. This procedure may be done to verify that the run is progressing as desired, to establish a steady-state flow field for a subsequent solute simulation, or to allow modification or inclusion of source/sink conditions starting at this time. The FEHM application shall provide this capability.

3.6.1 Write information needed for restart to output file

3.6.1.1 Introduction

All information needed to resume the computation at the final time of a simulation must first be written to an output file.

3.6.1.2 Inputs

The data needed in a subsequent restart are:

- final time of simulation and
- the pressures, temperatures, saturations, and concentrations (solute or noncondensible gas) at the final time step.

3.6.1.3 Processing

The processing shall write all needed data to the output restart file.

3.6.1.4 Outputs

The output shall be a restart file that can be subsequently read to restart a previous simulation or to start a new simulation with an established pressure, temperature, saturation, or concentration field.

3.6.2 Read information needed for restart from restart file

3.6.2.1 Introduction

The information needed to resume the simulation or start a new simulation with an existing pressure and temperature field shall be read in from the restart and primary input files.

DRAFT 4/97

3.6.2.2 Inputs

The required inputs are data needed in the restart, including all information written to the restart file in the previous simulation and new information such as source/sink flow rates or pressures for the restart found in the primary input files.

3.6.2.3 Processing

The processing shall read all data from the input files.

3.6.2.4 Outputs

The output shall be a data structure appropriate for resuming the computations, beginning at the final time and the conditions of the previous simulation.

3.6.3 Resume the calculation

3.6.3.1 Introduction

When all input information is read in, the calculation shall resume.

3.6.3.2 Inputs

The required inputs are all input parameter values from the restart and primary input files.

3.6.3.3 Processing

The processing shall perform the calculations as it would in the case of a starting time of 0 with the following exceptions:

- set the start time based on the value from the restart file, rather than 0;
- set the pressure field based on values from the restart file, rather than from the initial pressures;
- set the temperature field based on values from the restart file, rather than from the initial temperatures;
- set the saturations based on values from the restart file; and
- set the concentrations (solute or noncondensible gas) based on values from the restart file.

3.6.3.4 Outputs

The outputs shall be the computational results of a simulation starting at the time and conditions reached at the end of the previous execution of the program.

4.0 EXTERNAL INTERFACE REQUIREMENTS

4.1 User Interfaces

The names of input and output data files used by the programs shall be input interactively at the terminal when the programs are executed or shall be provided in optional configuration data files. All other user interfaces with the programs comprising the FEHM application shall be in the form of ASCII or binary data files. The data files employed by the application are:

DRAFT 4/97

- an optional configuration data file specifying the names of input and output data files used by each program of the application;
- one or more input data files containing the finite-element mesh information;
- one or more input data files that specify the control parameters, physical parameters, and the source/sink specifications for the model simulation;
- one or more output data files containing the results of the model simulation; and
- a restart data file that contains the information needed to resume a model computation at the termination time of a previous simulation (it becomes an input data file during the restart run).

4.2 Hardware Interfaces

N/A

4.3 Software Interfaces

4.3.1 Linear equation solver

In finite-element simulations, a set of residual equations is usually solved at each time step. When the equations are nonlinear, as in the FEHM application, the procedure normally employed is iterative, in which a set of linear algebraic equations is solved at each iteration until convergence of the nonlinear equations is achieved. In the FEHM application, the linear equation set at each iteration shall be solved with a reuse component, GZSOLVE, that provides a robust solution method for sparse systems of equations. The FEHM application shall supply the solver with the matrix equation set in a form suitable for the solver, and the solver shall return the solution vector for that iteration.

4.4 Communications Interfaces

N/A

5.0 PERFORMANCE REQUIREMENTS

N/A

6.0 DESIGN CONSTRAINTS

6.1 Standards Compliance

N/A

6.2 Hardware Limitations

N/A

7.0 SECURITY

N/A

8.0 OTHER REQUIREMENTS

8.1 Data Base

N/A

8.2 Operations

N/A

9.0 VALIDATION CRITERIA

Validation criteria shall be developed in Chapter III, “Verification and Validation Plan,” of this report.